

**In the Claims**

For the convenience of the Examiner all pending claims of the present Application are shown below whether an amendment has been made or not. Please amend the claims as follows:

1.       **(Cancelled)**
2.       **(Previously Presented)** A method for managing faults in a web service architecture comprising:
  - receiving a service request in a web service language, wherein the service request comprises invoking a service over a network;
  - translating the service request into a non-web service language;
  - executing the service request;
  - encountering an exception during the execution, wherein the execution comprises a fault preventing the fulfillment of the service request;
  - persisting the fault; and
  - providing a fault response.
3.       **(Previously Presented)** The method of Claim 2, wherein the service request is received from a service consumer, the service consumer coupled to the network.
4.       **(Original)** The method of Claim 3, wherein the fault response is provided to a fault service consumer, and wherein the fault service consumer is coupled to the network.
5.       **(Original)** The method of Claim 4, wherein the fault service consumer is the same as the service consumer.
6.       **(Previously Presented)** The method of Claim 2, wherein persisting the fault comprises labeling the fault with a unique identifier.
7.       **(Original)** The method of Claim 6, further comprising storing the fault in a database.

8. **(Original)** The method of Claim 7, further comprising storing multiple faults in the database, the storage comprising storing fault information.

9. **(Original)** The method of Claim 8, wherein providing a fault response comprises providing access to the database, the access operable to permit a user to track any fault stored in the database.

10. **(Original)** The method of Claim 8, wherein providing a fault response further comprises presenting the fault information in a console, the console operable to list the fault information stored in the database.

11. **(Previously Presented)** A system for managing faults in a service-oriented architecture comprising:

a service interface operable to:

receive a service request via a network, the service request received in a web service language; and

translate the service request into a non-web service language;

a service implementation coupled to the service interface, the service implementation operable to perform the service request and determine the source of any fault encountered in the performance;

a persistent store operable to persist any faults encountered in the performance; and

a fault service interface operable to transmit fault information.

12. **(Original)** The system of Claim 11, further comprising a fault service implementation coupled to the fault service interface, the fault service interface operable to retrieve the fault information from the persistent store.

13. **(Cancelled)**

14. **(Cancelled)**

15.     **(Original)** The system of Claim 12, wherein the fault service interface is further operable to receive fault status requests in a web service language and translate the fault status request into a non-web service language.

16.     **(Original)** The system of Claim 11, further comprising a service consumer, the service consumer coupled to the network and operable to transmit the service request to the service interface.

17.     **(Original)** The system of Claim 11, further comprising a fault service consumer, the fault service consumer coupled to the network and operable to receive the fault information from the fault service interface.

18.     **(Original)** The system of Claim 17, wherein the fault service consumer and the service consumer are the same consumer.

19.     **(Original)** The system of Claim 12, further comprising a fault network coupled to the network, the fault network operable to couple the service interface, service implementation, persistent store, and fault service interface.

20.     **(Original)** The system of Claim 11, wherein the persistent store is a database operable to store faults encountered during the performance.

21.     **(Original)** The system of Claim 11, wherein the service implementation is further operable to attach a unique identifier to each fault.

22.     **(Original)** The system of Claim 21, wherein the service implementation is further operable to direct the persistent store to store any faults according to the unique identifier.

23.     **(Original)** The system of Claim 20, wherein the database is further operable to store the faults in a web service language.

24. **(Original)** The system of Claim 12, wherein the fault service implementation is further operable to translate the fault information into a web service language.

25. **(Original)** The system of Claim 12, further comprising a console, the console operable to display fault information retrieved by the fault service implementation.

26. **(Previously Presented)** A system for managing faults in a web service architecture comprising:

a web service module coupled to a network and operable to manage service requests in a web service language, the web service module operable to:

receive a service request via a network, the service request received in the web service language; and

translate the service request into a non-web service language;

a diagnostic module operable to fulfill the service request and identify faults associated with the service request; and

a fault persistence module operable to store the faults in a persistent store.

27. **(Original)** The system of Claim 26, wherein the web service language is any protocol registered in the Universal Description Discovery and Integration registry.

28. **(Original)** The system of Claim 26, wherein the web service language is a remote procedure call.

29. **(Original)** The system of Claim 26, wherein the web service language is a HyperText Transfer Protocol.

30. **(Original)** The system of Claim 26, wherein the web service language is an application service interface.

31. **(Original)** The system of Claim 30, wherein the application service interface is Java message service.

32. **(Original)** The system of Claim 26, wherein the web service language is a protocol approved as a web service description language approved by the World Wide Web Consortium.

33. **(Original)** The system of Claim 26, wherein the persistent store is a database dedicated to the fault persistence module.

34. **(Original)** The system of Claim 26, wherein the web service module is further operable to receive service requests.

35. **(Original)** The system of Claim 26, further comprising a sub-network coupled to the web services module.

36. **(Original)** The system of Claim 35, further comprising at least one internal system, the at least one internal system coupled to the sub-network and operable to provide information required by the service request.

37. **(Original)** The system of Claim 36, wherein the diagnostic module is further operable to identify any faults caused by the at least one internal system.

38. **(Original)** The system of Claim 37, wherein the diagnostic module is further operable to communicate any faults to the fault persistence module.

39. **(Original)** The system of Claim 38, wherein the fault persistence module is further operable to label each fault with a unique identifier.

40. **(Original)** The system of Claim 39, wherein the fault persistence module is further operable to direct the persistent store to organize each fault by a unique identifier.

41. **(Original)** The system of Claim 26, wherein the web service module is further operable to receive a fault status request.

42. **(Original)** The system of Claim 41, wherein the fault status request is sent by a fault service consumer.

43. **(Original)** The system of Claim 42, wherein the fault service consumer is coupled to the sub-network.

44. **(Original)** The system of Claim 42, wherein the fault service consumer and the service consumer are the same.

45. **(Original)** A system for managing faults in a web services architecture comprising:

a system interface operable to receive a service request in a web services format, the system interface further operable to translate the service request into a non-web service format;

a service implementation operable to fulfill the service request, generate a fault report, and persist the fault, the persistence comprising storing the fault report in a persistent store, wherein generating a fault report comprises detecting a fault during the fulfillment of the service request, and persisting the fault comprises attaching a unique identifier to the fault report;

a fault service implementation operable to retrieve the fault report from the persistent store and translate the fault report into a web service format; and

a fault service interface operable to receive fault service requests and transmit a fault service response.